

Research and Applications

User stories as lightweight requirements for agile clinical decision support development

Vaishnavi Kannan,^{1,2} Mujeeb A Basit,^{1,3} Puneet Bajaj,³ Angela R Carrington,² Irma B Donahue,² Emily L Flahaven,¹ Richard Medford,^{1,3} Tsedey Melaku,⁴ Brett A Moran,^{3,4} Luis E Saldana,⁵ Duwayne L Willett,^{1,3} Josh E Youngblood,² and Seth M Toomay,^{1,6}

¹Clinical Informatics, University of Texas Southwestern Health System, Dallas, Texas, USA, ²Health System Information Resources Department, University of Texas Southwestern Medical Center, Dallas, Texas, USA, ³Department of Internal Medicine, University of Texas Southwestern Medical Center, Dallas, Texas, USA, ⁴Clinical Informatics, Parkland Health and Hospital System, Dallas, Texas, USA, ⁵Clinical Informatics, Texas Health Resources, Arlington, Texas, USA, and ⁶Department of Radiology, University of Texas Southwestern Medical Center, Dallas, Texas, USA

Corresponding Author: Duwayne L Willett, MD, MS, FAMIA, Clinical Informatics and Department of Internal Medicine, University of Texas Southwestern Medical Center, 5323 Harry Hines Blvd, Dallas, TX 75390-9047, USA; duwayne.willett@utsouthwestern.edu

Received 2 January 2019; Revised 17 May 2019; Editorial Decision 12 June 2019; Accepted 1 July 2019

ABSTRACT

Objective: We sought to demonstrate applicability of user stories, progressively elaborated by testable acceptance criteria, as lightweight requirements for agile development of clinical decision support (CDS).

Materials and Methods: User stories employed the template: As a [type of user], I want [some goal] so that [some reason]. From the “so that” section, CDS benefit measures were derived. Detailed acceptance criteria were elaborated through ensuing conversations. We estimated user story size with “story points,” and depicted multiple user stories with a use case diagram or feature breakdown structure. Large user stories were split to fit into 2-week iterations.

Results: One example user story was: As a rheumatologist, I want to be advised if my patient with rheumatoid arthritis is not on a disease-modifying anti-rheumatic drug (DMARD), so that they receive optimal therapy and can experience symptom improvement. This yielded a process measure (DMARD use), and an outcome measure (Clinical Disease Activity Index). Following implementation, the DMARD nonuse rate decreased from 3.7% to 1.4%. Patients with a high Clinical Disease Activity Index improved from 13.7% to 7%. For a thromboembolism prevention CDS project, diagrams organized multiple user stories.

Discussion: User stories written in the clinician’s voice aid CDS governance and lead naturally to measures of CDS effectiveness. Estimation of relative story size helps plan CDS delivery dates. User stories prove to be practical even on larger projects.

Conclusions: User stories concisely communicate the *who*, *what*, and *why* of a CDS request, and serve as lightweight requirements for agile development to meet the demand for increasingly diverse CDS.

Key words: clinical decision support, agile development, requirements, electronic health records, implementation

INTRODUCTION

Building the right thing (requirements analysis) and building the thing right (design) remain enduring challenges of new product development. Building the wrong thing—or incorrect requirements—proves particularly costly because all subsequent product development work is waste. Developing new clinical decision support (CDS) tools can be considered new product development, for which agile methods consistently outperform a traditional waterfall approach.^{1,2}

CDS requirements can be complex, understood differently by stakeholders, and prone to change following release into the real-world healthcare environment.^{3–5} Yet the need for new CDS likely will accelerate, with multiple drivers (Table 1).

To manage the volume and complexity of CDS requests more effectively, development teams are increasingly adopting agile methods.^{22–28} Agile methods employ iterative incremental development, with the goal of delivering value early and often in the form of working software developed each iteration, typically 1 to 4 weeks in length. These methods embrace change in response to feedback from actual users, to progressively improve a product's usefulness. To keep up with short iterations, agile development requires lightweight (low in overhead) requirements, while avoiding waste from incorrect requirements. Discovery of defects in existing CDS tools can prompt a call for more formal (or “heavier”) upfront requirements. Yet mandating increasingly comprehensive requirements in advance paradoxically can worsen delivery of high-quality fit-for-purpose tools. The benefits of rapid learning feedback cycles are lost, and the more extensively detailed initial requirements grow stale and inaccurate even before product release. Thus, the challenge remains of creating lightweight yet accurate CDS requirements. User stories have found favor in the agile community as a practical approach. In this work, we demonstrate the applicability of user stories, progressively elaborated by testable acceptance criteria, as lightweight requirements serving agile CDS development.

User stories as effective lightweight requirements for agile development

One reason why building the right thing proves to be so important can be seen by visualizing where product defects are created during the development life cycle, and the relative costs to remediate them. As shown in Figure 1, defects most frequently arise during the requirements and modeling phases, with decreasing frequency of defect creation during coding, testing, and release to production.²⁹ Conversely, the cost of fixing a defect increases exponentially along the development life cycle, with high costs to fix defects discovered after live in production. That is, it is relatively easy to get requirements wrong, and if not caught until late in the development cycle, remediating the error proves costly.

The high cost of inaccurate requirements spawned comprehensive requirements documents for signoff prior to design, followed by modeling and coding—the waterfall development approach. Perhaps surprisingly, this hasn't delivered the expected outcomes. Con-

versely, failure rates with this approach are high. Agile projects offer improvement (Supplementary Appendix A).^{1,30}

Using Agile methods, working software delivered at the end of each iteration enables adaptive feedback and refinement of requirements for the next iterations' deliverables, prioritized to deliver the most business value.^{2,31} Thus, detailed requirements, design, construction, and testing are all done within each iteration for the features delivered that cycle (Figure 2).²⁵

A rapid iteration pace necessitates lightweight, low-ceremony requirements, specification, and elaboration. From this challenge arose the practice of user story cards, now widely accepted in software development, and applicable to new product development in general.²⁵ The prevalent format proposed initially by Mike Cohn reads^{32,33}:

As a [type of user],
I want [some goal]
so that [some reason].

Simple to write and to read, this single sentence encapsulates the *who*, *what*, and *why* of a feature, providing a shared understanding of the desired end goal. The phrase “the 3 C's” describes the process, short for Card → Conversation → Confirmation. A user story, potentially handwritten initially on a 3 inch × 5 inch card, does not represent the final, complete requirements. Rather, developers and customers are encouraged to consider the story card as a promise to hold a conversation, during which more specific success-defining acceptance criteria will be elaborated. Last, a feature reaches confirmation when these criteria are met, and the feature has been delivered correctly and to the customer's satisfaction.

As stories get elaborated in more detail for slotting into a specific development iteration, not uncommonly they prove to be too large to complete fully in 1 iteration and need to be split into 2 or more child stories. To aid planning, developers can employ “story points” to estimate the relative size of stories.^{34–36} By quantitatively monitoring the total size of user stories successfully completed each iteration, teams determine their average story points completed per iteration (velocity), which can improve predictions of when future features will be delivered.

Role of user stories in overall agile development

User stories and acceptance criteria focus on the requirements phase of Figure 1, and on validation of the delivered final product. Additional agile methods apply to the remaining phases, including, among others, agile modeling,²⁹ model-driven development,^{37,38} test-driven development,³⁹ daily scrums,³¹ and automated regression testing.⁴⁰ Supplementary Appendix B includes an overview of a model-driven development process we've employed. Individual agile teams iteratively adapt their process as well as their product in response to end-iteration feedback and retrospectives (Supplementary Appendix C).³¹

Table 1. Drivers of increased volume and variety of CDS tool requests

Increased availability of digital health data: EHR data, genomic data, images, patient-generated data^{6,7}

Personalized and precision medicine, leading to more types of tailored recommendations^{8–12}

Shareable CDS,^{13,14} employing application programming interface standards such as Fast Healthcare Interoperability Resources and CDS Hooks^{15–17}

Patient-facing and clinician-facing CDS¹⁸

Local quality improvement projects and LHS initiatives applying a CDS component in the Evidence-to-Practice limb of the LHS cycle^{19–21}

CDS: clinical decision support; EHR: electronic health record; LHS: Learning Healthcare System.

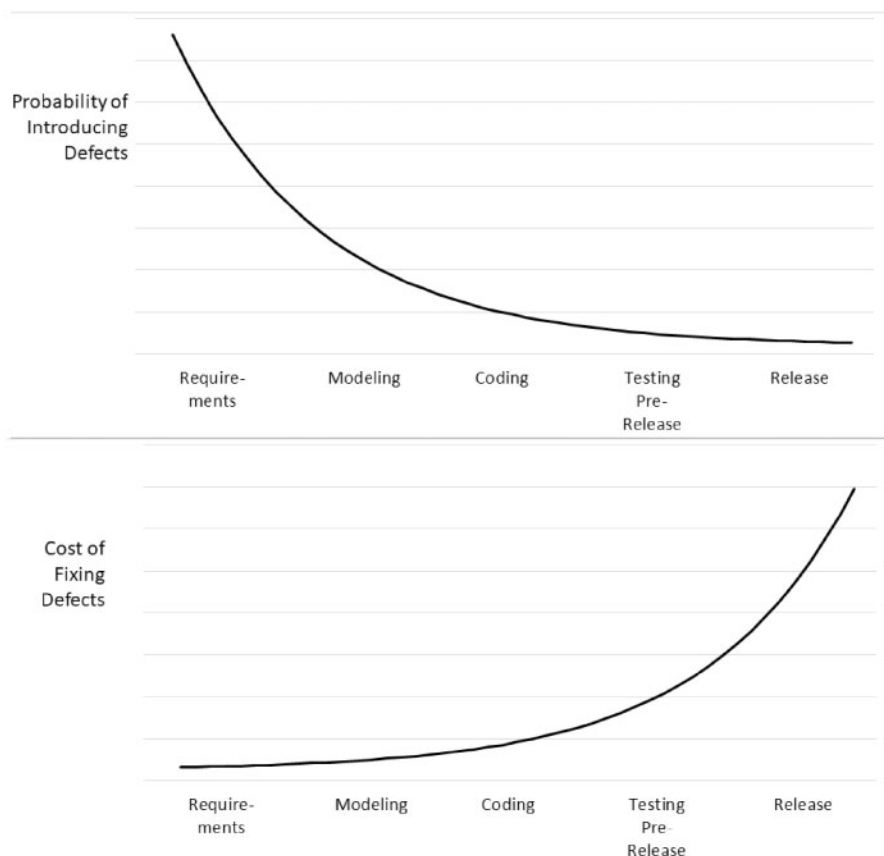


Figure 1. The decreasing probability of introducing defects during the development life cycle (top panel) and the rising costs of finding and fixing defects (bottom panel). Adapted with permission from Ambler.²⁹

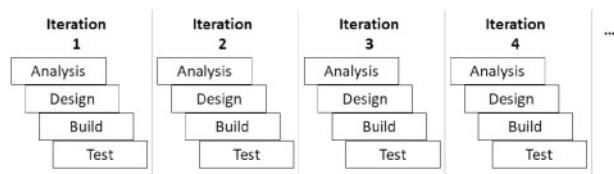


Figure 2. In agile development, the core activities of detailed analysis, design, build, and test are still done, but contemporaneously within one iteration, delivering a tested, production-ready feature that can be demonstrated at end iteration.

Applying user stories in agile CDS development

CDS tool development shares characteristics with new product development. The optimal design solution is not typically known up front: development can include learning from early versions, and incorporating feedback from actual end users to evolve the product. So, if we accept that new CDS development is a form of new product development, would applying user stories with acceptance criteria to CDS projects be similarly beneficial? As a succinct, readily understandable description, a user story could promote shared understanding of a newly proposed CDS tool among diverse clinical and nonclinical stakeholders, resolving a common challenge. Consistently writing the story in the voice of the person receiving the decision support could help to “do CDS with users, not to users.”⁵ The concluding part of the user story (so that [some reason]) can be particularly helpful in brainstorming what type of CDS might be most helpful in accomplishing the desired goal, and ensuring the intended effect is measurable.

While a user story begins to address the “5 rights of CDS” (right information, right person, right intervention format, right channel, right time in workflow),⁵ the remaining rights can be covered explicitly as acceptance criteria.

Techniques for splitting an overall software user story into child user stories or showing story relationships visually (eg, with a Story Map⁴¹) directly apply to large CDS projects and to the common scenario in which newly requested CDS tools are a component of a larger initiative.²⁶ Agile methods for estimating and planning with user stories³⁴ apply equally well to CDS tools. Story estimating potentially could improve prediction of CDS delivery dates during times of multiple contemporaneous CDS requests.

Objective

Demonstrate applicability of user stories, progressively elaborated by testable acceptance criteria, as lightweight requirements for agile development of CDS.

MATERIALS AND METHODS

Software and supplies

Drafting user stories employed readily available tools. Initially this involved 3x5” index cards, with acceptance criteria annotated on the card’s back. User story cards could then be placed on a wall for prioritizing and planning stories into upcoming iterations, and for monitoring status (to do, doing, done) within an iteration. Once scaled beyond a single team, user stories were written using elec-

Table 2. INVEST criteria for evaluating a user story

I	Independent	Ideally without inherent dependency on another user story
N	Negotiable	Not a specific contract, but leaves space for discussion
V	Valuable	Delivers something of value to stakeholders
E	Estimable	To a good approximation
S	Small	Small enough to fit within a single iteration if a near-term story. For future stories, not so big as to become impossible to plan and prioritize
T	Testable	In principle, even if there isn't a test for it yet

tronic collaboration tools (email, or a shared online notebook), and then submitted to our project request intake system (ServiceNow; ServiceNow, Inc., Santa Clara, CA). To manage large numbers of stories and handle story splitting and hierarchies, we employed a commercial agile project management tool (CA Central; CA Technologies, New York, NY). For automated acceptance testing and regression testing, we employed an open-source automated testing platform, FitNesse.^{42,43}

Setting

This work was done at the University of Texas Southwestern Health System (UT Southwestern), an academic medical center in Dallas, Texas, encompassing 2 hospitals and a large number of specialty ambulatory clinics all on a single electronic health record (EHR) instance (Epic; Epic Systems, Verona, WI). UT Southwestern shares a campus and long collaboration with Parkland Health and Hospital System and Children's Health, and is closely affiliated with Texas Health Resources as part of a clinically integrated network, Southwestern Health Resources. At UT Southwestern, requests for new or revised CDS tools come to a health system CDS Governance committee which includes ambulatory and inpatient physicians, nurses, pharmacists, laboratory and pathology representatives, clinical operations members, trainers, clinical informaticists, and members of the EHR and CDS analyst teams. The committee is co-chaired by the Chief Medical Informatics Officer and the manager of the CDS analyst team.

Procedures

Writing a user story

For writing user stories, we adopted the 3-line template from Cohn described previously.^{32,33}

We required that the "As a [type of user]" field be the recipient of the CDS, typically a clinical role, though it could be a patient role for patient-facing decision support.

In general, the team initially tried to describe the "I want [some goal]" entry in technology-agnostic terms: for instance, "I want to be advised that the patient..." rather than "I want a Best Practice Advisory to pop up that the patient..." By doing so, we hoped to encourage consideration of a wide array of possible methods for delivering decision support—not just interruptive pop-up alerts—and avoid premature lock-in to one specific method.

For clinician-facing CDS we sought to complete the "so that [some reason]" section with 1 or more of the following: (1) a direct benefit to the clinician (eg, streamlined work), (2) a benefit to their patient, or (3) a benefit to meeting a shared goal of the organization in which the clinician practices.

Most commonly, the story was written by an analyst after listening to the request for a new CDS tool, then reviewed with representative stakeholders for confirmation that the correct intent had been

captured, revising if necessary. Some stakeholders began submitting initial CDS requests as user stories.

Evolution of a user story was guided by comparing it with the tenets of the acronym INVEST (Table 2).^{32,44}

Some stories initially proved too large to schedule into a single iteration. Successfully splitting stories while following the INVEST criteria can be challenging. However, several heuristics now exist to spur thinking about how to split a story effectively. [Supplementary Appendix D](#) lists some examples of splitting CDS user stories employing the SPIDR mnemonic (spikes, paths, interfaces, data, and rules).⁴⁵

Deriving CDS effectiveness measures from a user story

From the "so that" section of the user story, we assessed how the intended benefit (to the clinician, patient, or organization) might be measured, both before and after CDS implementation. Both process measures and outcome measures were considered for inclusion.

Acceptance criteria, or conditions of satisfaction

Acceptance criteria were collected most frequently as simple bulleted lists. Delivery of any training materials needed for the CDS feature was included in the acceptance criteria. More detailed acceptance criteria could include truth tables or diagrams of expected external behavior (but not detailed internal design), or automated acceptance tests as executable requirements.³⁹ Automated acceptance tests, when included, were written as spreadsheet tables for vetting with clinicians and other stakeholders. These tables were then copied into an automated acceptance testing framework (FitNesse wiki) to query the EHR database or other databases for validation, as previously described.^{42,46,47} Addition of acceptance criteria could occur at our CDS Governance committee meeting, where for instance the type of CDS or needed training method might be decided. Elaboration of more detailed acceptance criteria occurred most often in conversations among the requestor, CDS analyst team member(s), and a clinical informatician.

Estimating and planning with user stories

Estimating the size or bigness of individual user stories employed the story point method popularized by Cohn.^{34,35} In this method, the relative size of a story (amount of work needed) is gauged relative to other user stories, using a numerical variant of the "T-shirt size" estimation method (eg, S, M, L, XL). Our team adopted the modified Fibonacci series commonly employed for story point sizes: 1, 2, 3, 5, 8, 13, 20, 40, 100.³⁴ To get the process started, 1 story point was initially equated to roughly 1 "ideal developer day." This represents the amount of work a single typical developer could accomplish, given no competing tasks and approximately 5 hours available for pure development effort.

To improve internal consistency of estimates, the practice of Planning Poker was introduced.^{34,36} In this modified Delphi technique, a story and its acceptance criteria are presented to the entire team. Each team member silently estimates the story's size and places a card with that number face down. Everyone then simultaneously turns over their cards and compares numbers. The persons with the highest and lowest estimates are invited to share their rationale, to encourage dialog about simpler ways to approach the work or overlooked work needed to complete the story. After brief discussion, the team members then re-estimate individually, and again simultaneously reveal their estimates. Typically, 2-3 rounds result in convergence. If not, after 3 rounds, the team lead selected the best consensus number.

At the end of each iteration, the team's productivity was measured as the sum of story points for user stories entirely completed and able to be demonstrated at end iteration. No credit was given for partially completed stories. Over a span of several iterations, the average number of successfully completed story points per iteration was calculated as team velocity (units = story points/iteration). Team velocity was used in combination with story point size estimates of upcoming work to predict completion dates.^{34,35}

Scaling user stories for larger projects

For scaling user stories to larger projects, we used 1 or more of the following methods. For some projects, our team used a spreadsheet to display multiple user stories concisely, with 1 row per user story and 3 column headings—1 for each component of a user story (As a..., I want..., so that...). Additional useful columns included a short title of each user story, the size estimate in story points, and a story's priority. Spreadsheet tools then allowed for facile sorting, filtering, or prioritizing.

For larger projects, our team opted for a use case diagram, depicting each user story as an oval use case (within a rectangle denoting the overall project scope).⁴⁸ Each oval was connected by a line to a stick figure for the role(s) interacting with that user story. The end result was a single-page diagram summarizing the value to be created from the perspective of potential users of the system.⁴⁹

On other large projects, we employed a feature breakdown structure (FBS) to emphasize organization of the user stories to be delivered. Resembling an organizational chart, a FBS—also known as a work breakdown structure in traditional project management—conveys a whole-parts decomposition of a project into its various components.

RESULTS

Writing user stories

The following example CDS user story (part of a specialty rheumatoid arthritis patient registry project) keeps to the who-what-why format:

As a rheumatologist,

I want to be advised if my patient with rheumatoid arthritis (RA) is not on a Disease Modifying Anti-Rheumatic Drug (DMARD), so that they receive optimal therapy for their RA, and can experience improvement in their symptoms and quality of life.

We strove always to write the user story in the voice of the person engaging with the CDS tool. This at times meant re-writing initial user stories to achieve the perspective of the clinician (or patient) user, as in the following example:

Initial "nonuser story":

As Pharmacy and Therapeutics Committee co-chairs, we want the EHR to prevent physicians from ordering IV hydromorphone so that we won't run out, which we are on track to do in 7 days despite sending 2 email memos to all medical staff.

Rewritten user story:

As a provider ordering IV hydromorphone (Dilaudid), I want to be advised of the national shortage and prompted to order an alternative instead, so that our nearly depleted stock is preserved for those patients of mine and my colleagues who truly cannot be given the alternatives of intravenous (IV) morphine or oral hydromorphone.

Acceptance criteria, or conditions of satisfaction

For the rheumatoid arthritis CDS advisory user story, the acceptance criteria were expressed as a bulleted list, as follows:

- Alert within EHR to prompt ordering of a DMARD on patients with rheumatoid arthritis
- Alert displays for patients with Rheumatoid Arthritis on their Problem List in the EHR (based on Systematized Nomenclature of Medicine)
- Alert does not display if a DMARD medication is already on their active medication list
- Alert allows designation of reasons why the patient is an exception, and should not be on a DMARD currently
- Tip sheet to train on use of alert

For the hydromorphone shortage advisory, the acceptance criteria were:

- Display when hydromorphone IV injectable is being ordered (list provided by pharmacy service)
- Does not display if creatinine clearance < 60 mL/min
- Ability to remove triggering hydromorphone injectable order within best practice advisory (BPA)
- Ability to add alternative orders (morphine IV or hydromorphone orally) from BPA
- Equianalgesic dose table included on BPA user interface (provided by pharmacy)
- Acknowledgement reasons include "true allergy to morphine," "pre-existing hypotension," and "BPA fired inappropriately"

The user interface for the completed hydromorphone advisory is shown in [Figure 3](#):

Measures of CDS effectiveness derived from user stories

Because the "so that" section of a CDS user story lists the most compelling benefit(s) to be achieved, measures of CDS effect are derived naturally. For the hydromorphone shortage advisory, this section read in part "so that our nearly depleted stock is preserved for those patients of mine and my colleagues who truly cannot be given the alternatives." A follow-up email from the P&T Committee co-chair reported on the results as follows:

Wanted to provide you an update on hydromorphone usage. Since the BPA was moved to production, the number of hydromorphone doses dispensed from the Pharmacy has decreased from approximately 400–450 to 100–110 doses per day. Based on current usage, we have enough to last for 2 to 3 weeks.

Ultimately, low quantities of hydromorphone doses began arriving, and the stock never ran out completely.

For the RA advisory, the "so that" section translated directly to a process measure (use of a DMARD), and an outcome measure (Clinical Disease Activity Index), categorized as remission (best), low, medium, or high, as shown in [Table 3](#).

Following implementation, the DMARD Compliance Defect rate decreased from 3.7% to 1.4% over 6 months. The percentage of patients with a high (unfavorable) Clinical Disease Activity Index improved from 13.7% to 10.1% over the same period, and currently is 7%.⁵⁰

Estimating and planning with user stories

Specific examples of estimating the relative size of user stories with story points, and their allocation to 2-week iterations, are shown in [Table 4](#).

BestPractice Advisory - Zzztest,Dora

Important (1)

Due to a national shortage of IV hydromorphone (Dilaudid), our stock is approaching depletion. Please order an alternative (IV morphine or PO hydromorphone) below, unless this patient is truly allergic to morphine (i.e. anaphylaxis) or has pre-existing hypotension.

Equianalgesic Dose (mg)		
Hydromorphone IV	Morphine IV	Hydromorphone Oral
1.5 mg	10 mg	7.5 mg
1.0 mg	6 mg	5 mg
0.5 mg	3 mg	3 mg
0.25 mg	1.6 mg	1.5 mg
0.1 mg	0.7 mg	1 mg

Remove the following orders?

HYDROmorphine (DILAUDID) injection DOSE:
intraVENOUS, Starting Today, Until Discontinued If IV and PO pain medication ordered for same pain level, begin with oral medication. Give IV if not tolerating oral medication, patient NPO, or oral medication is ineffective.

Apply the following?

morphine injection 2 mg/1 mL

morphine injection 4 mg/1 mL

morphine injection 10 mg/ 1 mL

HYDROmorphine (DILAUDID) tablet

Acknowledge Reason

© 2019 Epic Systems Corporation. Used with permission.

Figure 3. User interface for the completed hydromorphone advisory, delivered ready for production release 48 hours after the initial request.

Table 3. Process and outcome measures of CDS benefit derived from the user story

"So that" clause of user story	Measure Type	Measure
"[my patients] receive optimal therapy for their RA"	Process	DMARD Compliance Rate: % of RA patients eligible for a DMARD who were prescribed one Defect rate = 1 – DMARD compliance rate
"experience improvement in their symptoms and quality of life"	Outcome	Clinical Disease Activity Index—a combination of patient-reported outcomes and clinician physical assessment

CDS: clinical decision support; DMARD: disease-modifying anti-rheumatic drug; RA: rheumatoid arthritis.

These comprise a subset of stories from a larger venous thromboembolism (VTE) prevention CDS redesign project (see also [Figures 4 and 5](#)).

Scaling user stories for larger projects

A use case diagram depicted component user stories for the VTE CDS project, emphasizing what behaviors each user role will experience from the system ([Figure 4](#)).

As a complementary view, user stories were shown as a FBS, decomposing the whole initiative into its component deliverables ([Figure 5](#)).

From either diagram, user stories were prioritized for slotting into one of the upcoming development iterations (or into subsequent

multi-iteration releases), including splitting user stories further if needed to fit within our 2-week iteration time boxes ([Supplementary Appendix D](#)).

DISCUSSION

Principal findings

User stories with acceptance criteria proved practical as initial light-weight requirements for CDS tools, succinctly specifying the who, what, and why of a request in the voice of the clinician recipient. The "so that" clause of the user story led naturally to measures of CDS effectiveness, which can include both process and outcome

Table 4. Sample user stories with relative size estimation in story points for a VTE prevention project

User story ID	Story name	User story	Story points	Assigned iteration
US 1591	VTE Risk SmartForm for Data Entry	As a Clinician, I want to review and update a VTE Risk Form so that I can determine my patient's risk of VTE (risk category), and ensure proper prophylaxis	8	FY 2016 04/08/2016–04/21/2016
US 1600	BPA to Prompt Ordering VTE Prophylaxis on Admission	As an Inpatient MD/APP, I want to be prompted to Order VTE prophylaxis on admission (if not done through an Admission Order Set), so that I remember to place my patient on VTE prophylaxis.	5	FY 2016 06/03/2016–06/16/2016
US 1599	VTE Prophylaxis Dynamic Order Group in Admit Order Sets	As an Inpatient MD/APP, I want to view only risk-appropriate VTE Prophylaxis options in Admission Order Sets so that I can ensure my patient is getting optimal VTE prophylaxis.	13	FY 2016 08/12/2016–08/25/2016

APP, advanced practice provider; BPA: best practice advisory; FY: fiscal year; VTE: venous thromboembolism.

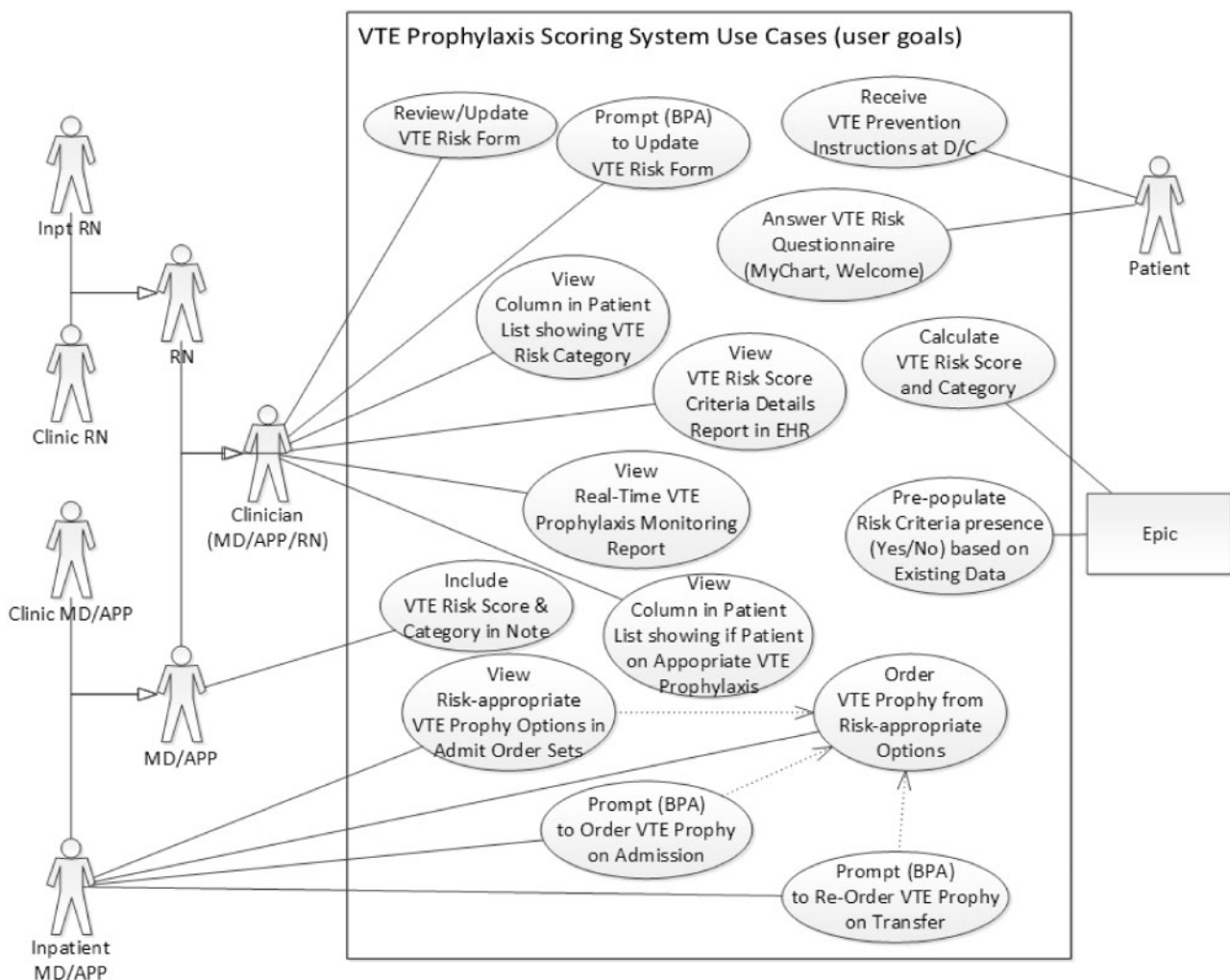


Figure 4. Use case diagram for a venous thromboembolism (VTE) Risk Scoring and Prophylaxis CDS Redesign project. APP: advanced practice provider; BPA: best practice advisory; RN: registered nurse.

measures. Estimation of relative size as story points enabled forecasting of CDS delivery dates. Larger projects could be managed through a combination of use case diagrams, FBS diagrams, and story splitting.

Practical implications

Governance of CDS

Governance of CDS requests poses challenges at all stages, from requests to development and through roll-out.⁵¹ Challenges arise

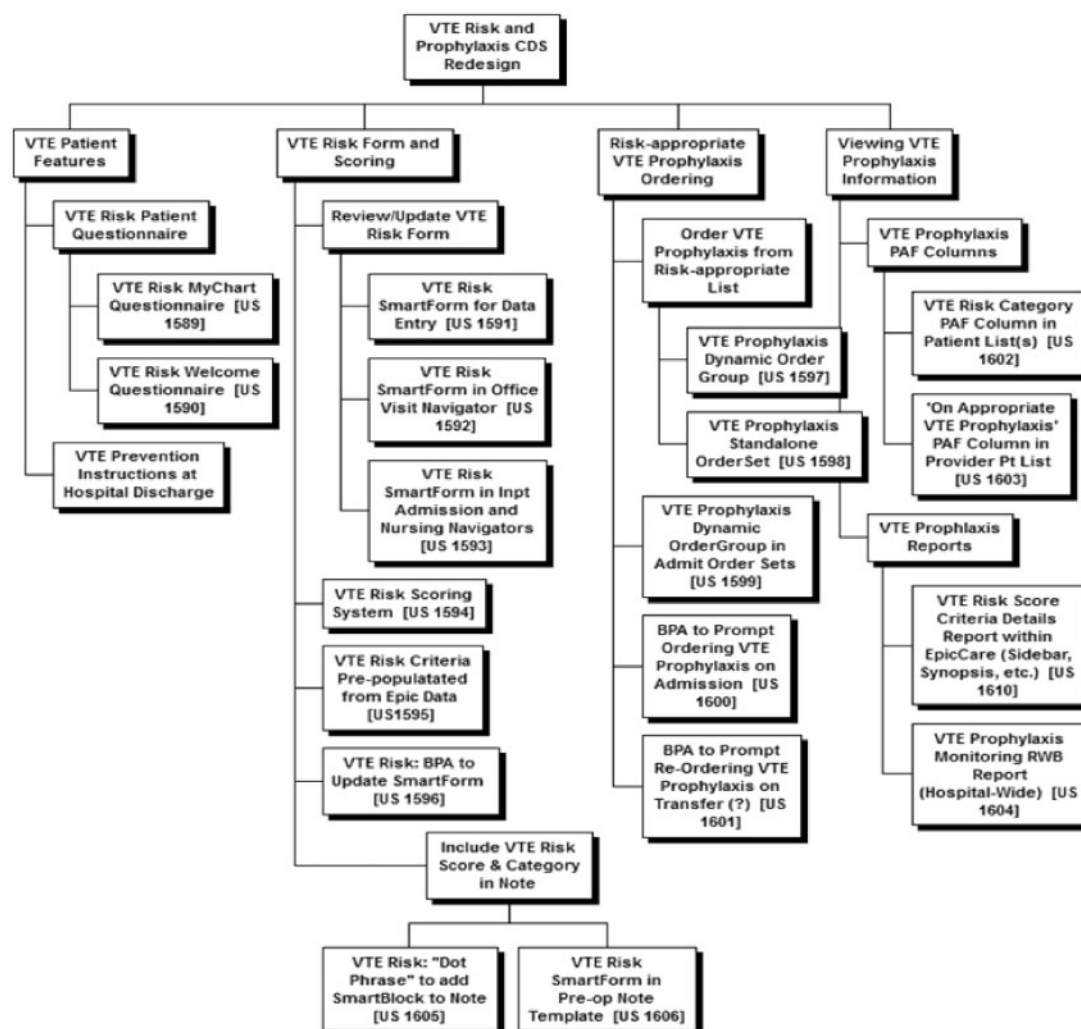


Figure 5. Feature breakdown structure for a venous thromboembolism (VTE) Risk Scoring and Prophylaxis CDS Redesign project.

from the multiplicity of stakeholders and intervention options, and limited development resources require prioritization. Lightweight yet informative, user stories provide a valuable starting point for governance discussions on selecting an appropriate CDS method for each story and on prioritization. User stories likewise help avoid unintentional evolution toward a “CDS Prevention Program” if overly detailed CDS design specifications are mandated with the initial request. While well-meaning, such an approach risks skewing prioritization of CDS selected for development based primarily on requestor resources or persistence, rather than on intrinsic merits of the CDS idea and potential for patient benefit.

Proposed measures of CDS effectiveness derived from the user story’s “so that” clause can be reviewed during CDS project evaluation. Such measures enable postlaunch assessment of CDS benefits realization, which can help to (a) justify ongoing investment in CDS by documenting favorable impact, and (b) mitigate negative impacts of CDS by turning off tools failing to achieve their objective. Additionally, the user story phrase readily translates to a Release Notes format (“As a __, you can now __, so that __”) suitable for concisely communicating benefits of the new CDS tool to clinicians during roll-out. Organizations can realize these benefits of user stories for their CDS governance process even if not yet adopting agile methods for CDS development.

Creating a single overall use case diagram for a broad clinical topic (eg, “VTE prophylaxis,” or “palliative care”) supports CDS governance by placing any new requests for CDS tools in conjunction with existing CDS for that topic and depicting the recipient of each CDS feature. This helps to identify potentially conflicting CDS, and also to rationalize the CDS behavior experienced by clinicians.

CDS defect reduction

Defects in released CDS tools are surprisingly common.^{52–54} User stories may help reduce defects by iteratively assuring we “build the right thing” (Figure 1). When elaborating user stories with acceptance criteria, expanding the use of automated acceptance tests would align CDS development more closely with modern development practices for high-quality software. Automated acceptance test-driven development provides an additional lever for achieving high-reliability CDS tools.^{55,56}

CDS team productivity and sustainable pace

Measuring a team’s effective productivity by delivery of features actually employed for valuable purposes leads to the sobering realization that, by some estimates of custom software development, roughly half of features developed fail to be adopted by end

users.^{57,58} Effective productivity in terms of delivering business value can thus be sharply increased—even doubled—by avoiding building features never used. Engaging stakeholders in capturing requests as user stories and elaborating acceptance criteria helps achieve shared understanding of the desired feature early on, reducing project abandonment. Adoption of agile methods by UT Southwestern's EHR team was accompanied by a 75% increase in features delivered to production per month, and yet a 17% decrease in customer-reported defects.⁵⁰ As experience grows with story point estimation and team velocity tracking, assignment of CDS work into iterations becomes increasingly predictable. Team members can work at a reasonable, sustainable pace, while gaining the satisfaction of seeing what they have built used in production to help clinical care.

Adoption of agile delivery of new features more generally

For organizations electing to adopt agile methods, user stories can positively reinforce the transition. The ease of writing user stories helps combat 2 tendencies that counteract agile development's benefits. One tendency is to progressively add more upfront ceremony and detail in response to product or process problems, as previously discussed. In lean methodology terms, time spent writing detailed requirements in advance that don't match what's ultimately built, or for something never used, represents waste. It is not that agile development rejects detail. Rather, precision of specifications for a delivered product often increases under agile development—it is just not all done at once.^{59,60} By iteratively elaborating requirements, teams can progressively expand the detail and correctness of requirements for what is actually delivered, and minimize waste.

Adopting user stories also helps counteract a second tendency: to return to a traditional waterfall development mindset within the context of iterations. This typically takes the form of adding an upfront requirements iteration, then 1 or more analysis iterations, then a series of build iterations, and a final testing iteration. Some initial architectural modeling certainly benefits larger initiatives (in what's been called a planning "iteration zero").²⁹ But, in general, finely detailed analysis and design, building, and testing of a single user story are best completed inside a single iteration (Figure 2), thereby delivering working, tested product features at the end of each iteration. Doing so maximizes the adaptive learning crucial to achieving optimal benefits from agile delivery.

Limitations

User stories constitute just one approach to requirements analysis (though in practice we've found all our CDS requests can initially be expressed as user stories). Adopting user stories does not preclude employing other CDS models and frameworks, such as the CDS 5 Rights.⁵ User stories with acceptance criteria describe the destination (or definition of success) when developing a CDS tool, but are not design and thus do not provide the full route to that destination. For design, other models (such as decision trees, object models, state diagrams, and user interface story boards) prove useful, which can profitably make use of agile modeling principles and practices.^{23,29,48} User-centered design offers promising methods for an effective design process meeting the goals of a user story.^{61–64}

Additionally, user stories do not address by themselves the important need for knowledge management. Knowledge management systems can keep track of CDS artifacts deployed in an organization, along with their relationships to various initiatives and to each other.⁵¹ Our EHR contains a metadata management capability,

which we have found to be useful for tracking CDS review cycles, changes, and associations with organizational or governmental initiatives. For persistent documentation of design, we favor the efficiency of test-driven development, where acceptance criteria for a user story are elaborated as automated acceptance tests. Such automated tests then serve 4 purposes: as requirements before construction, verification upon construction, regression tests once in production, and living documentation of business rules and design for later reference.^{39,55}

No mechanism may currently exist to measure the "so that" clause of the user story for gauging CDS benefit. If so, whether to include creating that data collection capability as one of the story's acceptance criteria becomes a scoping question. If included, then the story's size estimate (in story points) grows to cover the work to meet all acceptance criteria. Stories too large to fit in a single iteration can be split by acceptance criteria or by the dimensions in SPIDR (Supplementary Appendix D). Creating the data capture method would be a strong candidate for splitting into 1 (or more) child stories.

CONCLUSION

By concisely capturing the *who*, *what*, and *why* of a CDS request, user stories focus attention on the decision support value to be provided and to whom. In providing lightweight initial requirements, a user story serves agile development well, facilitating conversations during governance decision making and leading to progressive elaboration with acceptance criteria for shared understanding. Estimating story sizes with story points can improve planning and predictability of delivery. Ongoing (ideally continuous) monitoring of CDS in production provides a way for CDS governance to ensure the "why" portion of the user story continues to be met. With the coming acceleration of requests for more personalized and interoperable CDS, user stories offer a welcome nimble method for helping ensure—even during rapid-cycle development—that we "build the right thing."

FUNDING

Dr. Willett received support from the National Center for Advancing Translational Sciences of the National Institutes of Health under award number UL1TR001105. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

AUTHOR CONTRIBUTIONS

Conception and/or design of agile approaches to CDS requirements (listed alphabetically): MAB, ARC, VK, TM, BAM, LES, SMT, DLW, JEY. Creating user stories and/or acceptance criteria for the rheumatoid arthritis CDS project: PB, ARC, VK, DLW, JEY. Creating user stories and/or acceptance criteria for the hydromorphone CDS project: VK, DLW. Creating user stories and/or acceptance criteria for the VTE CDS project: MAB, ARC, IRD, ELF, VK, TM, DLW, JEY. Analyzing and/or interpretation of data for specific user stories and/or on productivity: MAB, PB, ARC, IRD, ELF, VK, RM, TM, SMT, DLW, JEY. Drafting the manuscript text and/or figures: MAB, ARC, VK, SMT, DLW (primary writer: DLW). Critically revising the manuscript: MAB, PB, ARC, IRD, ELF, VK, RM, TM, BAM, LES, SMT, JEY. Final approval of version to be published,

and agreement to be accountable for all aspects of work: MAB, PB, ARC, IRD, ELF, VK, RM, TM, BAM, LES, SMT, DLW, JEY.

SUPPLEMENTARY MATERIAL

Supplementary material is available at *Journal of the American Medical Informatics Association* online.

ACKNOWLEDGMENTS

Ashley Chen, clinical decision support analyst at UT Southwestern, developed the hydromorphone CDS advisory, working in conjunction with William Tharpe, clinical pharmacy manager at UT Southwestern University Hospitals. Carol Croft and Michael Burton served as physician leaders for the VTE prophylaxis redesign project and participated extensively in developing user story requirements. Mallory Willett critiqued the article draft and contributed valuable edits. We thank Jason Fish, Deepa Bhat, Ki Lai and their teams for indispensable work on UT Southwestern's chronic disease registry analytics, including the Rheumatoid Arthritis Registry project. We also thank the following UT Southwestern Information Resources leaders for their consistent support in advancing clinical decision support activities: Marc Milstein, Mark Rauschuber, Kathryn Flores, Jimmie Glorioso, and Scott Minnerly.

CONFLICT OF INTEREST STATEMENT

None declared.

REFERENCES

- Hastie SW. Standish Group 2015 Chaos Report-Q&A with Jennifer Lynch. 2015. <https://www.infoq.com/articles/standish-chaos-2015>. Accessed October 4, 2015.
- Larman C. *Agile and Iterative Development: A Manager's Guide*. London: Pearson Education; 2003.
- Sittig DF, Wright A, Osheroff JA, et al. Grand challenges in clinical decision support. *J Biomed Inform* 2008; 41 (2): 387–92.
- Middleton B, Sittig DF, Wright A. Clinical decision support: a 25 year retrospective and a 25 year vision. *Yearb Med Inform* 2016; 25(Suppl 1): S103–16.
- Osheroff JA, Teich J, Levick D, et al. *Improving Outcomes with Clinical Decision Support: An Implementer's Guide*. Chicago: Health care Information and Management Systems Society; 2012.
- Dolley S. Big data's role in precision public health. *Front Public Health* 2018; 6: 68.
- Kruse CS, Goswamy R, Raval Y, Marawi S. Challenges and opportunities of big data in health care: a systematic review. *JMIR Med Inform* 2016; 4 (4): e38.
- Frey LJ, Bernstam EV, Denny JC. Precision medicine informatics. *J Am Med Inform Assoc* 2016; 23 (4): 668–70.
- Caraballo PJ, Bielinski SJ, St Sauver JL, Weinshilboum RM. Electronic medical record-integrated pharmacogenomics and related clinical decision support concepts. *Clin Pharmacol Ther* 2017; 102 (2): 254–64.
- Freimuth RR, Formea CM, Hoffman JM, Matey E, Peterson JF, Boyce RD. Implementing genomic clinical decision support for drug-based precision medicine. *CPT Pharmacometrics Syst Pharmacol* 2017; 6 (3): 153–5.
- Sitapati A, Kim H, Berkovich B, et al. Integrated precision medicine: the role of electronic health records in delivering personalized treatment. *Syst Biol Med* 2017; 9 (3): e1378.
- Warner JL, Rieth MJ, Mandl KD, et al. SMART precision cancer medicine: a FHIR-based app to provide genomic information at the point of care. *J Am Med Inform Assoc* 2016; 23 (4): 701–10.
- Marco-Ruiz L, Pedrinaci C, Maldonado JA, Panziera L, Chen R, Bellika JG. Publication, discovery and interoperability of clinical decision support systems: a linked data approach. *J Biomed Inform* 2016; 62: 243–64.
- Agency for Healthcare Research and Quality. Welcome to CDS connect. <https://cds.ahrq.gov/cdsconnect>. Accessed November 29, 2018.
- HL7. FHIR overview. <https://www.hl7.org/fhir/overview.html>. Accessed October 29, 2017.
- Benson T, Grieve G. *Principles of Health Interoperability: SNOMED CT, HL7, and FHIR*. 3rd ed. London: Springer; 2016.
- CDS Hooks. 2018. <http://cds-hooks.org/>. Accessed October 29, 2017.
- Payne HE, Lister C, West JH, Bernhardt JM. Behavioral functionality of mobile apps in health interventions: a systematic review of the literature. *JMIR mHealth uHealth* 2015; 3 (1): e20.
- Friedman CP, Wong AK, Blumenthal D. Achieving a nationwide learning health system. *Sci Transl Med* 2010; 2 (57): 57cm29.
- Maddox TM, Albert NM, Borden WB, et al. The learning healthcare system and cardiovascular care: a scientific statement from the American Heart Association. *Circulation* 2017; 135 (14): e826–57.
- Richesson RL, Green BB, Laws R, et al. Pragmatic (trial) informatics: a perspective from the NIH Health Care Systems Research Collaboratory. *J Am Med Inform Assoc* 2017; 24 (5): 996–1001.
- Banos O, Villalonga C, Garcia R, et al. Design, implementation and validation of a novel open framework for agile development of mobile health applications. *Biomed Eng Online* 2015; 14 Suppl 2: S6.
- Kannan V, Willett DL. Agile clinical decision support development. Agile alliance experience reports; 2016. <https://www.agilealliance.org/resources/experience-reports/agile-clinical-decision-support-developments/>. Accessed March 28, 2016.
- Kannan V, Basit MA, Willett DL. Agile clinical decision support development. In: *AMIA 2017 Annual Symposium*. Washington, DC: American Medical Informatics Association; 2017: 103–4.
- Kannan V, Basit MA, Youngblood JE, et al. Agile co-development for clinical adoption and adaptation of innovative technologies. In: *IEEE Healthcare Innovations and Point of Care Technologies (HI-POCT)*; 2017: 56–9.
- Kannan V, Fish JS, Mutz JM, et al. Rapid development of specialty population registries and quality measures from electronic health record data: an agile framework. *Methods Inf Med* 2017; 56: e74–83.
- Robertson GE, Wakefield EC, Cohn JR, O'Brien T, Ziegler SD, Fardell EJ. The development of delta: using agile to develop a decision aid for pediatric oncology clinical trial enrollment. *JMIR Res Protoc* 2018; 7 (5): e119.
- Shattuck D, Haile LT, Simmons RG. Lessons from the dot contraceptive efficacy study: analysis of the use of agile development to improve recruitment and enrollment for mHealth research. *JMIR mHealth uHealth* 2018; 6 (4): e99.
- Ambler SW. *Agile Modeling: Effective Practices for Extreme Programming and the Unified Process*. Hoboken, NJ: Wiley; 2002.
- Agile vs. waterfall: survey shows agile is now the norm. 2017. <https://techbeacon.com/survey-agile-new-norm>. Accessed June 26, 2018.
- Rubin KS. *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. Boston: Addison-Wesley Professional; 2012.
- Cohn M. *User Stories Applied: For Agile Software Development*. Boston: Addison Wesley Longman; 2004.
- Cohn M. User stories. 2018. <https://www.mountaingoatsoftware.com/agile/user-stories>. Accessed May 2, 2018.
- Cohn M. *Agile Estimating and Planning*. Upper Saddle River, NJ: Prentice Hall; 2005.
- Agile Estimating and Planning by Mike Cohn. 2018. <https://www.mountaingoatsoftware.com/books/agile-estimating-and-planning>. Accessed April 19, 2018.
- Usman M, Mendes E, Weidt F, Britto R. Effort estimation in agile software development: a systematic literature review. In: *PROMISE '14 Pro*

- ceedings of the 10th International Conference on Predictive Models in Software Engineering; 2014.
37. Rosenberg D, Stephens M. *Use Case Driven Object Modeling with UML: Theory and Practice*. 2nd ed. New York: Apress; 2013.
 38. Larman C. *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*. 3rd ed. Upper Saddle River, NJ: Prentice Hall; 2004.
 39. Adzic G. *Specification by Example: How Successful Teams Deliver the Right Software*. Shelter Island, NY: Manning; 2011.
 40. Gregory J, Crispin L. *More Agile Testing: Learning Journeys for the Whole Team*. Boston: Addison-Wesley Professional; 2014.
 41. Patton J, Economy P. *User Story Mapping: Discover the Whole Story, Build the Right Product*. Sebastopol, CA: O'Reilly Media, Inc.; 2014.
 42. Adzic G. *Test Driven NET Development with FitNesse*. London: Neuri Limited; 2008.
 43. FitNesse: The fully integrated standalone wiki and acceptance testing framework. <http://fitnesse.org/>. Accessed June 26, 2018.
 44. Agile Alliance. INVEST. 2018. <https://www.agilealliance.org/glossary/invest>. Accessed November 27, 2018.
 45. Lattenkamp K. SPIDR—five simple techniques for a perfectly split user story. 2017. <https://blogs.itemis.com/en/spidr-five-simple-techniques-for-a-perfectly-split-user-story>. Accessed November 25, 2018.
 46. Nikolov Y. DbFit-automated open source database testing. 2016. <http://www.methodsandtools.com/tools/dbfit.php>. Accessed August 1, 2017.
 47. Keshvani N, Kulkarni N, Parks CJ, et al. Test-driven development of clinical decision support for atrial fibrillation. In: *AMIA Clinical Informatics Conference*. Scottsdale, AZ: American Medical Informatics Association; 2018.
 48. Kannan V, Fish J, Willett DL. Agile model driven development of electronic health record-based specialty population registries. In: 2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI). Las Vegas, NV, USA: IEEE; 2016: 465–68.
 49. Kannan V, Fish JS, Mutz JM, et al. Rapid development of specialty population registries and quality measures from electronic health record data: supplementary material. *Methods Inf Med* 2017; 56: e74–83.
 50. Youngblood JE, Fish JS, Willett DL. UT Southwestern Davies Enterprise Award EHR-based specialty registries. 2017. <https://www.himss.org/library/ut-southwestern-davies-enterprise-award>. Accessed December 8, 2018.
 51. Wright A, Sittig DF, Ash JS, et al. Governance for clinical decision support: case studies and recommended practices from leading institutions. *J Am Med Inform Assoc* 2011; 18 (2): 187–94.
 52. Wright A, Hickman TT, McEvoy D, et al. Analysis of clinical decision support system malfunctions: a case series and survey. *J Am Med Inform Assoc* 2016; 23 (6): 1068–76.
 53. Kassakian SZ, Yackel TR, Gorman PN, Dorr DA. Clinical decision support malfunctions in a commercial electronic health record. *Appl Clin Inform* 2017; 8 (3): 910–23.
 54. Ray S, McEvoy DS, Aaron S, Hickman T-T, Wright A. Using statistical anomaly detection models to find clinical decision support malfunctions. *J Am Med Inform Assoc* 2018; 25 (7): 862–71.
 55. Basit MA, Baldwin KL, Kannan V, et al. Agile acceptance test-driven development of clinical decision support advisories: feasibility of using open source software. *JMIR Med Inform* 2018; 6 (2): e23.
 56. Sittig DF, Singh H. Toward more proactive approaches to safety in the electronic health record era. *Jt Comm J Qual Patient Saf* 2017; 43 (10): 540–7.
 57. Johnson JR. It's your job. In: *Third International Conference on Extreme Programming*. Alghero, Italy; 2002.
 58. Standish. *Exceeding Value*. Boston: The Standish Group International; 2014.
 59. Ambler S. The disciplined agile (DA) framework: a foundation for business agility. 2017. <http://www.disciplinedagiledelivery.com/>.
 60. Ambler S, Lines M. An executive's guide to disciplined agile: winning the race to business agility. CreateSpace; 2017.
 61. Horsky J, Schiff GD, Johnston D, Mercincavage L, Bell D, Middleton B. Interface design principles for usable decision support: a targeted review of best practices for clinical prescribing interventions. *J Biomed Inform* 2012; 45 (6): 1202–16.
 62. Horsky J, Phansalkar S, Desai A, Bell D, Middleton B. Design of decision support interventions for medication prescribing. *Int J Med Inform* 2013; 82 (6): 492–503.
 63. Jeffery AD, Novak LL, Kennedy B, Dietrich MS, Mion LC. Participatory design of probability-based decision support tools for in-hospital nurses. *J Am Med Inform Assoc* 2017; 24 (6): 1102–10.
 64. Miller K, Mosby D, Capan M, et al. Interface, information, interaction: a narrative review of design and functional requirements for clinical decision support. *J Am Med Inform Assoc* 2018; 25 (5): 585–92.